Office Of The Secretary Of Defense (OSD)
Deputy Director Of Defense Research & Engineering
Deputy Under Secretary Of Defense (Science & Technology)
Small Business Technology Transfer Research (STTR)
FY 2006 Program Description

Introduction

The Deputy Under Secretary of Defense (Science & Technology) STTR Program is sponsoring two information systems technology themes in this solicitation.

The Army, Navy, and Air Force are participating in the OSD program this year. The service laboratories act as our OSD Agent in the management and execution of the contracts with small businesses. The service laboratories, often referred to as a DoD Component acting on behalf of the OSD, invite small business firms to submit proposals under this Small Business Technology Transfer Research (STTR) Program solicitation. In order to participate in the OSD STTR Program this year, all potential proposers should register on the DoD SBIR/STTR website as soon as you can, and should follow the instruction for electronic submittal of proposals. It is required that all bidders submit their proposal cover sheet, company commercialization report and their firm's technical and cost proposal form electronically through the DoD SBIR/STTR Proposal Submission Website at http://www.dodsbir.net/submission. If you experience problems submitting your proposal, call the help desk (toll free) at 1-866-724-7457. You must include a Company Commercialization Report as part of each proposal you submit; however, it does not count against the proposal page limit. Please note that improper handling of this form may result in the proposal being substantially delayed. Information provided may have a direct impact on the review of the proposal. The DoD SBIR/STTR Proposal Submission Website allows your company to come in any time (prior to the proposal submission deadline) to edit your Cover Sheets, Technical and Cost Proposal and Company Commercialization Report.

We <u>WILL NOT</u> accept any proposals that are not submitted through the on-line submission site. The submission site does not limit the overall file size for each electronic proposal, there is only a page limit. However, file uploads may take a great deal of time depending on your file size and your internet server connection speed. If you wish to upload a very large file, it is highly recommended that you submit prior to the deadline submittal date, as the last day is heavily trafficked. You are responsible for performing a virus check on each technical proposal file to be uploaded electronically. The detection of a virus on any submission may be cause for the rejection of the proposal. We will not accept e-mail submissions.

Firms with strong research and development capabilities in science or engineering in any of the topic areas described in this section and with the ability to commercialize the results are encouraged to participate. Subject to availability of funds, the DUSD(S&T) STTR Program will support high quality research and development proposals of innovative concepts to solve the listed defense-related scientific or engineering problems, especially those concepts that also have high potential for commercialization in the private sector. Objectives of the DUSD(S&T) STTR Program include stimulating technological innovation, strengthening the role of small business in meeting DoD research and development needs, fostering and encouraging participation by minority and disadvantaged persons in technological innovation, and increasing the commercial application of DoD-supported research and development results. The guidelines presented in the solicitation incorporate and exploit the flexibility of the SBA Policy Directive to encourage proposals based on scientific and technical approaches most likely to yield results important to DoD and the private sector.

Description of the OSD STTR Three Phase Program

Phase I is to determine, insofar as possible, the scientific or technical merit and feasibility of ideas submitted under the STTR Program and will typically be one half-person year effort over a period not to exceed six months, with a dollar value up to $100,000. We plan to fund 3 Phase I contracts, on average, and downselect to one Phase II contract per topic. This is assuming that the proposals are sufficient in quality to fund this many. Proposals should concentrate on that research and development which will significantly contribute to proving the scientific and technical feasibility of the proposed effort, the successful completion of which is a prerequisite for further DoD support in Phase II. The measure of Phase I success includes technical performance toward the topic objectives and evaluations of the extent to which Phase II results would have the potential to yield a product or process of continuing importance to DoD and the private sector, in accordance with Section 4.3.

Subsequent Phase II awards will be made to firms on the basis of results from the Phase I effort and the scientific and technical merit of the Phase II proposal in addressing the goals and objectives described in the topic. Phase II awards will typically cover 2 to 5 person-years of effort over a period generally not to exceed 24 months (subject to negotiation). Phase II is the principal research and development effort and is expected to produce a well defined deliverable prototype or process. A more comprehensive proposal will be required for Phase II.

Under Phase III, the DoD may award non-STTR funded follow-on contracts for products or processes, which meet the component mission needs. This solicitation is designed, in part, to encourage the conversion of federally sponsored research and development innovation into private sector applications. The small business is expected to use non-federal capital to pursue private sector applications of the research and development.

This solicitation is for Phase I proposals only. Any proposal submitted under prior STTR solicitations will not be considered under this solicitation; however, offerors who were not awarded a contract in response to a particular topic under prior STTR solicitations are free to update or modify and submit the same or modified proposal if it is responsive to any of the topics listed in this section.

For Phase II, no separate solicitation will be issued and no unsolicited proposals will be accepted. Only those firms that were awarded Phase I contracts, and have successfully completed their Phase I efforts, will be invited to submit a Phase II proposal. Invitations to submit Phase II proposals will be released at or before the end of the Phase I period of performance. The decision to invite a Phase II proposal will be made based upon the success of the Phase I contract to meet the technical goals of the topic, as well as the overall merit based upon the criteria in section 4.3. DoD is not obligated to make any awards under Phase I, II, or III. DoD is not responsible for any money expended by the proposer before award of any contract. For specifics regarding the evaluation and award of Phase I or II contracts, please read the front section of this solicitation very carefully. Every Phase II proposal will be reviewed for overall merit based upon the criteria in section 4.3 of this solicitation, repeated below:

a. The soundness, technical merit, and innovation of the proposed approach and its incremental progress toward topic or subtopic solution.
b. The qualifications of the proposed principal/key investigators, supporting staff, and consultants. Qualifications include not only the ability to perform the research and development but also the ability to commercialize the results.
c. The potential for commercial (defense and private sector) application and the benefits expected to accrue from this commercialization.

In addition, the OSD STTR Program has a *Phase II Plus* Program, which provides matching STTR funds to expand an existing Phase II that attracts investment funds from a DoD acquisition program or Private sector investments. ***Phase II Plus*** allows for an existing Phase II OSD STTR effort to be extended for up to one year to perform additional research and development. ***Phase II Plus*** matching funds will be provided on a one-for-one basis up to a <u>maximum $250,000</u> of STTR funds. All ***Phase II Plus*** awards are subject to acceptance, review, and selection of candidate projects, are subject to availability of funding, and successful negotiation and award of a ***Phase II Plus*** contract modification.

The Fast Track provisions in section 4.0 of this solicitation apply as follows. Under the Fast Track policy, STTR projects that attract matching cash from an outside investor for their Phase II effort have an opportunity to receive interim funding between Phases I and II, to be evaluated for Phase II under an expedited process, and to be selected for Phase II award provided they meet or exceed the technical thresholds and have met their Phase I technical goals, as discussed in Section 4.5. Under the Fast Track Program, a company submits a Fast Track application, including statement of work and cost estimate, within 120 to 180 days of the award of a Phase I contract (see the Fast Track Application Form on www.dodsbir.net/submission). Also submitted at this time is a commitment of third party funding for Phase II. Subsequently, the company must submit its Phase I Final Report and its Phase II proposal no later than 210 days after the effective date of Phase I, and must certify, within 45 days of being selected for Phase II award, that all matching funds have been transferred to the company. For projects that qualify for the Fast Track (as discussed in Section 4.5), DoD will evaluate the Phase II proposals in an expedited manner in accordance with the above criteria, and may select these proposals for Phase II award provided: (1) they meet or exceed selection criteria (a) and (b) above and (2) the project has substantially met its Phase I technical goals (and assuming budgetary and other programmatic factors are met, as discussed in Section 4.1). Fast Track proposals, having attracted matching cash from an outside investor, presumptively meet criterion (c). However, selection and award of a Fast Track proposal is not mandated and DoD retains the discretion not to select or fund any Fast Track proposal.

Follow-On Funding

In addition to supporting scientific and engineering research and development, another important goal of the program is conversion of DoD-supported research and development into commercial products. Proposers are encouraged to obtain a contingent commitment for private follow-on funding prior to Phase II where it is felt that the research and development has commercial potential in the private sector. Proposers who feel that their research and development have the potential to meet private sector market needs, in addition to meeting DoD objectives, are encouraged to obtain non-federal follow-on funding for Phase III to pursue private sector development. The commitment should be obtained during the course of Phase I performance. This commitment may be contingent upon the DoD supported development meeting some specific technical objectives in Phase II which if met, would justify non-federal funding to pursue further development for commercial purposes in Phase III. The recipient will be permitted to obtain commercial rights to any invention made in either Phase I or Phase II, subject to the patent policies stated elsewhere in this solicitation.

Contact with DoD

General informational questions pertaining to proposal instructions contained in this solicitation should be directed to the topic authors and point of contact identified in the topic description section. Proposals should be electronically submitted. Oral communications with DoD personnel regarding the technical content of this solicitation during the pre-solicitation phase are allowed, however, proposal evaluation is conducted only on the written submittal. Oral communications during the pre-solicitation period should be considered informal, and will not be factored into the selection for award of contracts. Oral communications subsequent to the pre-solicitation period, during the Phase I proposal preparation periods are <u>prohibited</u> for reasons of competitive fairness. Refer to the front section of the solicitation for the exact dates.

Proposal Submission

Proposals shall be submitted in response to a specific topic identified in the following topic description sections. The topics listed are the only topics for which proposals will be accepted. Scientific and technical information assistance may be requested by using the SBIR/STTR Interactive Technical Information System (SITIS).

It is required that all bidders submit their proposal cover sheet, company commercialization report and their firm's technical and cost proposal form electronically through the DoD SBIR/STTR Proposal Submission Website at http://www.dodsbir.net/submission. If you experience problems submitting your proposal, call the help desk (toll free) at 866-724-7457. You must include a Company Commercialization Report as part of each proposal you submit; however, it does not count against the proposal page limit. Please note that improper handling of this form may result in the proposal being substantially delayed. Information provided may have a direct impact on the review of the proposal. The proposal submission website allows your company to come in any time (prior to the proposal submission deadline) to edit your Cover Sheets, Technical and Cost Proposal and Company Commercialization Report. We **<u>WILL NOT</u> accept any proposals which are not submitted through the on-line submission site.** The submission site does not limit the overall file size for each electronic proposal, only the number of pages are limited. However, file uploads may take a great deal of time depending on your file size and your internet server connection speed. You are responsible for performing a virus check on each technical proposal file to be uploaded electronically. The detection of a virus on any submission may be cause for the rejection of the proposal. We will not accept e-mail submissions.

The following is a summary of the technology areas, which are followed by the topics.

Enabling Network Centric Operations Technology Area

As envisioned, the Global Information Grid (GIG) will connect the roughly 3 million computers, 100,000 LANs, 100 long-distance networks, and a multitude of wireless networks and devices in support of all DoD, national security, and related intelligence community missions and functions. It will provide the joint warfighter with a single, end-to-end information system capability, built on a secure, robust network-centric environment, allowing users to post and access shared data and applications regardless of their location – while inhibiting or denying an adversary's ability to do the same. The future vision is a converged heterogeneous enterprise capable of protecting content of different sensitivities. However, the GIG construct, while highly desirable from a functionality viewpoint, presents serious challenges from a security perspective. DoD's unprecedented enterprise vision for future information operations must simultaneously address protecting and defending its critical information and information technology systems by ensuring availability, integrity, authentication, confidentiality and non-repudiation; and by providing security management and operations that incorporate the requisite protection, detection, and quick reaction capabilities.

The converged, decentralized vision of the future network requires a parallel adoption of a decentralized trust paradigm. Degrees of trust and robustness hitherto provided by enclave isolation and separation must be distributed to across the networks down to the tactical edge devices. With increasing joint, allied and coalition operations, dynamic and secure collaboration and data sharing across security domains is a critical capability.

DoD is making significant investments in ensuring the security of net-centric operations of the GIG. However, the scope of the challenges and the dynamics of the information technology industry provide multiple opportunities for new and innovative security solutions. In particular new technology solutions are needed for supporting the edge users who must operate across multiple domains and communications paths, on less hardened networks, to reach other tactical mission players, and to access protected core information systems and data warehouses.

The Network Centric Operations technology area addresses the broad technical challenges of developing the technology to ensure fundamental trust of mission critical systems and information (aka Trusting the Edge), to provide the basis for security management (aka Security Management Infrastructure), to enable our customers to transit and/or tolerate a hostile environment to conduct business, including communicate securely, outside the traditional secure enclave (aka Mobility), to enable safe, trusted information exchange (aka Assured Information Sharing), and to provide a seamless, integrated situational awareness capability and rapid, automated protection response capability (aka Enterprise Health: Situational Awareness and Response) across different network security domains. The objectives are to provide new or novel technological/operational capabilities for Defense systems. Potential concepts include:

- Methods and approaches for more robust user and device authentication. Within the new distributed architecture crossing multiple security levels, continuous, two-way authentication is needed to provide mutual trust between the edge devices, end-users, communities, and enterprise servers.
- Innovative methodologies to provide data authenticity and integrity. To implement the consumer-driven "need to share" and "post before process" paradigms, the binding of metadata and security attribute to establish pedigree, integrity, and releasability of data are needed.
- New concepts of trusted platforms operating across multiple security domains while maintaining the functionality users demand. Tools and techniques for policy-based security to enable cross-domain file transfers, email, and collaboration is integral to future concepts of operations.
- New security policy management tools and techniques. Today's technology does not support the needed capability to dynamically change security policies in a trusted and assured manner across the enterprise. Tools are needed to allow security policies to change over time while maintaining the appropriate degree of security and accountability commensurate with the mission. As an example new methods are needed to describe, analyze and understand the combined or composed security policies in as networks are linked.
- Advanced network architectures to provide for prioritized, fault-tolerant operations. The convergence of multiple independent operational networks requires that the GIG maintain a robust core capability while under operational stress, physical and logical attack. Fault-tolerant designs are needed to dynamically

prioritize mission critical activities. Networks must be made self-healing and self-forming to as great a degree as possible.

- New concepts and techniques for situational awareness. An integrated security management and situation awareness framework is needed that utilizes all network devices to provide a coherent, global operational picture. The ability to detect a sophisticated, patient adversary and to provide tools for subsequent risk management analysis is needed.

The Network Centric Operations Technology topics are:

OSD06-NC1	Effective Development, Configuration, and Control of Intrusion Detection Systems (Army)
OSD06-NC2	Automatic Generation of Robust Network Intrusion Detection Signatures (Army)
OSD06-NC3	Data Base Security for Army Mobile Environments (Army)
OSD06-NC4	Kernel-mode Software Protection Vulnerability Assessment and Rootkit Reverse Engineering Tool Development (Air Force)
OSD06-NC5	Deobfuscating tools for the validation and verification of tamper-proofed software (Air Force)

Software Producibility Technology Area

DoD's reliance on software and software-enabled technologies to maintain superiority on the battlefield is increasing exponentially. As we demand larger and more complex systems, we rapidly exceed the capabilities of our industry to develop and systems engineer individual systems and the interdependencies between those systems. Our appetite for software has exceeded our ability to produce it. Much of the mission functionality demanded from programs such as F/A-22, JSF, Future Combat System, and many others is embodied in large, complex software systems.

Shortcomings in software development often lead to schedule slippage, cost growth, and mission compromises. These shortcomings can frequently be traced to underpowered software development technologies not up to the task of developing the scale and complexity of software needed. Despite the large role of the commercial sector in advancing software technology, there are many key aspects of complex, distributed, robust systems crucial to DoD that may not be addressed directly by commercial technology efforts, as our experience over the past decade shows. DoD needs to reinvigorate an investment in producing software because mechanisms to address the size and complexity of systems and our need for reliable, real-time, interoperable mission-critical software are not being provided by commercial industry.

Technical goals are to meet and ensure mission-critical requirements; control complexities; enable system evolution; ensure seamless interoperability; model behavior and performance; and improve performance against cost and schedule expectations. The research program should identify, develop, or transition promising software technologies involving: specification of complex requirements; correct-by-construction software development; composable and customizable frameworks; high-confidence system software and middleware; system Information Technology architectures for interoperability and network-centric environments; technologies for testing, verification, and validation; and, modeling and metrics. Additional research areas include: observability of software-based systems; approaches to specifying and ensuring desired behavioral characteristics in software and software based systems (performance, reliability, timeliness, security etc...); managing complexity of capabilities-driven and functionally-interdependent systems of systems; modeling complex software and networked systems behavior; and system architectures for interoperability and net-centric environments. Some potential solutions to the software problem may require establishing a fundamental theory and science as the basis for developing effective tools and approaches.

The Software Producibility Technology topics are:

OSD06-SP1 Error-Handling Paths and Policies Analysis (Navy)
OSD06-SP2 Security Escorts for Not-Yet-Trusted Software (Air Force)
OSD06-SP3 Assessing Interoperability Through Cross-Domain Protocol Compatibility Analysis (Army)
OSD06-SP4 Software System Reliability Analysis (Army)

OSD STTR 06 Topic Index

OSD STTR 06 Topic Descriptions

OSD06-NC1 TITLE: Effective Development, Configuration, and Control of Intrusion Detection Systems

TECHNOLOGY AREAS: Information Systems

OBJECTIVE: The objective of this STTR is to research and develop a high confidence software framework that supports the rapid on demand development and deployment of dynamic intrusion detection systems and the implementation of a dynamic and real-time control infrastructure for managing intrusion detection and response.

DESCRIPTION: In recent years, networks have evolved from a mere means of communication to a ubiquitous computational infrastructure. Networks have become larger, faster, and highly dynamic. In particular, the Internet, the world-wide TCP/IP network, has become a mission-critical infrastructure for governments, companies, financial institutions, and millions of everyday users. The surveillance and security monitoring of the network infrastructure is mostly performed using Intrusion Detection Systems (IDSs) deployed throughout a protected network. The intrusion detection community has developed a number of different systems that perform intrusion detection in particular domains (e.g., hosts or networks), in specific environments (e.g., Windows NT or Solaris), and at different levels of abstraction (e.g., kernel-level tools and alert correlation systems). Unfortunately, these systems suffer from two main limitations: 1) they are developed ad hoc, domains and/or environments specific; 2) deploying, configuring, and managing a large number of heterogeneous IDSs is a complex, expensive, and error-prone activity.

Today's networks are not only heterogeneous; they are also dynamic. Therefore, intrusion detection systems need to support mechanisms to dynamically change/update their configuration and to create new sensors as the security state of the protected system evolves. To address the complexity of managing intrusion detection infrastructures, what is needed is a software framework for the rapid development of intrusion detection sensors dynamically (on demand) and a control infrastructure for managing these systems in real time response. The intrusion detection sensors would be built by dynamically composing domain-specific components with a domain-independent runtime. An IDS would have the ability to reconfigure its behavior dynamically. The reconfiguration functionality would be supported by a component model and by a control infrastructure. The final product of the software framework would be a highly-configurable, well-integrated intrusion detection infrastructure that can respond in real time to attacks.

Under this STTR, a well-performing and high-confidence software framework that supports the rapid development and deployment of new intrusion detection sensors and the implementation of an intrusion detection infrastructure that advances the state-of-the-art of building, configuring, controlling, and managing intrusion detection systems. The intrusion detection framework needs to support:

1. Be able to dynamically configure a running IDS so that new event streams can be used as input for the security analysis:

New attacks may have manifestations in event streams that are not currently analyzed by a specific IDS. The infrastructure must provide the basic mechanisms to reconfigure system for security analysis.

2. Be able to dynamically include new signatures at execution time:

Most existing intrusion detection systems are initialized with a set of signatures at startup time, and updating the signature set requires stopping the IDS, adding new signatures, and then restarting execution. The framework must be capable of adding new signatures, changing signature parameters, and activating and deactivating signatures in real-time while the IDS is running.

3. Be able to dynamically control responses and to associate a response with intermediate steps in an attack:

The configuration of responses in existing intrusion detection systems is relatively static. In most cases it is possible to choose only from a specific subset of possible responses. The infrastructure must provide mechanisms to reconfigure system response at run-time.

4. Support the explicit modeling of the dependencies among the modules composing an intrusion detection system:

It must be possible to automatically identify the steps that are necessary to perform a reconfiguration of the deployed sensing infrastructure.

PHASE I:

a. Research and develop configuration primitives, algorithms, and communication protocols for the design of the software infrastructure. The system should support dynamic system generation and reconfiguration discussed in the last section. The system framework must include a domain-independent attack modeling language and a domain-independent event analysis engine.
b. Demonstrate that the proposed algorithms and protocols can meet the needs.

PHASE II:
a. Develop a working system (software implementation) for building, configuring and controlling IDSs.
b. Carry out experiments on fielding IDSs on a network and on dynamically changing their configurations.
c. Demonstrate the advantages of this approach by comparing against existing tools and techniques for detecting intrusions on a wide-area network.

PHASE III/DUAL USE APPLICATIONS: The surveillance and security monitoring of the network infrastructure is important for both the military and commercial communities. Although the information security policies in each community are usually different, the detection engine should address any range of policies through the use of multiple models of attacks as required. The system developed needs to be adaptable to both military and civilian computer systems.

KEYWORDS: IDS, dynamic reconfiguration, real-time update, intrusion response, network monitoring and surveillance

OSD06-NC2 TITLE: Automatic Generation of Robust Network Intrusion Detection Signatures

TECHNOLOGY AREAS: Information Systems

OBJECTIVE: The objective of this STTR is to research and develop automatic generation of highly accurate network intrusion detection signatures.

DESCRIPTION: Network intrusion detection systems (NIDS) provides information to network security analysts on the characteristics of malicious attacks and intrusions in a network, and, in the best case, enable the security infrastructure to be adapted (manually) to current conditions. Unfortunately, it is well know that NIDS suffer from two important problems. First, they commonly have high false alarm rates (both positive and negative) that can significantly reduce their effectiveness. Second, the signatures used to detect new attacks and variants of existing attacks must be crafted by hand after the new attack vector has been recognized through some other mechanism. This STTR aims at research and development of a system that can automatically generate signatures for network intrusion detection systems.

The general task of developing signatures for network intrusion detection systems is challenging for several reasons. The most significant of these are, (i) the inherent variability in benign traffic leads to high false alarm rates and (ii) adversaries can examine signature sets and with simple modifications can develop variants of attacks that evade detection [1,2]. To address these issues, signatures should be as attack specific as possible to reduce the potential for false alarms, but at the same time, they should be as general as possible to limit the effectiveness of simple modifications of attacks to elude detection. As mentioned above, new attack are typically recognized through mechanisms such as volume-based anomaly detectors. Recent studies have developed methods for coupling volume-based identification with longest common substring identification to automatically generate signatures to identify worm outbreaks [3-5]. However, volume-based detection is also easy to avoid which leads to the possibility of new attacks remaining undetected for extended periods, thus signatures from these systems are less effective for general task of intrusion detection. Honeynets are becoming more widely used as a mechanism for detecting new types of attacks [6-9]. Another set of studies has developed methods for harnessing the capabilities of honeynets to automatically generate signatures for NIDS [10-11].

Under this STTR, an implementation of an automatic network intrusion detection signature generation system must advance the state of the art of in the following areas:

1. Signatures to detect broad classes of attacks:

The signatures generated by the system must not be limited to detection of a single attack vector (eg. worms that scan at high rates) or a single service (eg. HTTP). Instead signatures that detect a broad range of attacks on the most popular services must be addressed.

2. Signatures that result in low false alarm rates:

The technology developed must advance the accuracy of network attack detection in the presence of code obfuscation and highly variable benign traffic. Improvement in detection accuracy must be verifiable via benchmark testing

3. Real time or near real time generation:

The system must generate signatures in real time or near real time. Real time signature generation is the ultimate goal, since near real time may not be able to address some types of fast scanning worms.

4. Adaptability for site specific deployment:

The signatures generated by this system must be able to be used in NIDS that are commonly deployed in the Internet (both commercial and open source). The signatures should also be able to be inspected by security administrators so that they can be augmented with specific details if necessary.

PHASE I:

a. Research and develop automatic signature generation algorithm(s) that can accurately detect a broad class of attacks and are resilient to standard methods of obfuscation.

b. Demonstrate that the proposed algorithm(s) can accurately detect a broad class of attacks and are resilient to standard obfuscation methods.

PHASE II:

a. Develop a working system that automatically generates signatures that can be deployed in a NIDS and tested on live traffic.

b. Carry out comprehensive benchmarking experiments using synthetic traffic generators (both benign and malicious) with the capability of varying attacks and obfuscation levels.

c. Demonstrate the advantages of this approach by comparing against existing tools and techniques for detecting network attack activity. Benchmarking test needs to be designed and developed to carry out the detection accuracy evaluation and understand the speed of generation.

PHASE III/DUAL USE APPLICATIONS: Network intrusion detection is a critical capability in both the military and commercial sectors. Although the network security policies in each community are likely to be different, the signatures that are used in intrusion detection systems should address the widest possible range of attacks as well as site-specific policies as required. The developed signature generation technology must be able to be adapted for use in both commercial and open source intrusion detection systems, especially those used on military and other government networks, and in commercial networks that are common targets for attack (eg. the financial sector). The developed system should be marketed as a standalone product or can be licensed to a third party.

REFERENCES:

[1] S. Rubin, S. Jha, and B. Miller, “Automatic generation and analysis of NIDS attacks,” In the Annual Computer Security Applications Conference, December 2004.

[2] M. Handley, C. Kreibich, and V. Paxson, “Network Intrusion Detection: Evasion, Traffic Normalization, and End-to-End Protocol Semantics,” In Proceedings of the USENIX Security Symposium 2001.

[3] S. Singh, C. Estan, G. Varghese, and S. Savage “Automated Worm Fingerprinting”, In Proceedings of USENIX OSDI, December 2004.

[4] H-A. Kim, and B. Karp, B., “Autograph: Toward Automated, Distributed Worm Signature Detection”, In Proceedings of the 13th USENIX Security Symposium, August, 2004.

[5] J. Newsome, B. Karp, and D. Song, Polygraph: “Automatically Generating Signatures for Polymorphic Worms”, In Proceedings of the IEEE Symposium on Security and Privacy, May, 2005.

[6] The Honeynet Project, http://www.honeynet.org, 2005.

[7] V. Yegneswaran, P. Barford, and D. Plonka, “On the Design and Use of Internet Sinks for Network Abuse Monitoring”, In Proceedings of Symposium on Recent Advances in Intrusion Detection (RAID), September, 2004.

[8] M. Bailey, E. Cooke, F. Jahanian, J. Nazario, and D. Waston, “The Internet Motion Sensor: A Distributed Blackhole Monitoring System”, In Proceedings of Network and Distributed System Security Symposium (NDSS), February, 2005.

[9] M. Vrable, J. Ma, J. Chen, D. Moore, E. Vandekieft, A. Snoeren, G. Voelker, and S. Savage, "Scalability, Fidelity and Containment in the Potemkin Virtual Honeyfarm", In
Proceedings of the ACM Symposium on Operating System Principles (SOSP), October 2005.
[10] C. Keribich, and J. Crowcroft, "Honeycomb: Creating Intrusion Detection Signatures Using Honeypots", In Proceedings of the Second Workshop on Hot Topics in Networks (Hotnets II), November, 2003.
[11] V. Yegneswaran, J. Giffin, P. Barford, and S. Jha, "An Architecture for Generating Semantic-aware Signatures", In proceedings of USENIX Security Symposium, August, 2005.

KEYWORDS: IDS system, automatic signature generation

OSD06-NC3 TITLE: Data Base Security for Army Mobile Environments

TECHNOLOGY AREAS: Information Systems

OBJECTIVE: Perform research into Data Base Security mechanisms for Army Mobile Ad-Hoc Networks (MANETS) that are typical of the Army's Future Combat System and Warfighter Information Network- Tactical environments. The security solutions formulated would be extremely useful to both the commercial and military worlds. Note that it is anticipated that the security solutions formulated would also be extremely beneficial in the Homeland Defense application by protecting critical computer network infrastructures.

DESCRIPTION: In both the commercial and military world, Data Base security is being recognized as a major emerging problem. It is vital to protect computers and computer networks from hacker and foreign power threats. There are a number of commercially available Data Base Security products that can effectively operate in a static environment, but provide little support for a highly dynamic environment such as MANET. This type of environment can be characterized by:

a. highly dynamic networks with mobile nodes and infrastructure (routing, security, configuration)
b. typically, no concentration points where traffic can be analyzed
c. network addresses do not always reflect location (physical or hierarchical)
d. cannot rely on centralized network or security services
e. intermittent connectivity caused by mobility/ noise
f. normal user behavior not easily characterized due to mobility of networks and dynamic tactical missions
g. forward deployed nodes are susceptible to enemy capture
h. energy, processing, storage constraints
i. bandwidth constraints.

This research will investigate new and innovative approaches for Data Base Security solutions within this highly distributed environment.

PHASE I: Perform a study of possible computer and computer network Data Base Security solutions for MANETS. Contractor should also perform a study of what is needed for this project in the encryption area, beyond what is currently commercially available for Personal Digital Assistants (PDAs). A set of alternatives would then be presented to the government. The contractor and the government would make a joint decision on the most promising techniques to pursue in Phase II.

PHASE II: The most promising techniques emerging from the Phase I effort would be further developed and modeled. A performance description or specification would be developed. A prototype software working model will be delivered.

PHASE III: Military use would include Data Base Security solutions for soldiers who are assigned Wireless Devices in MANET environments. These devices are becoming more prevalent in the military. Commercial uses would include personnel who are assigned Handheld Wireless Devices in such diverse industries as banking, electric power utilities, telephone systems, police and emergency civilian personnel, etc. Note that it is anticipated that the Data Base Security solutions formulated would also be extremely beneficial in the Homeland Defense application by protecting critical computer network infrastructures against outsiders attempting to break into the network.

REFERENCES:
www.symantec.com Database Security
www.database.about.com/od/security Database Security Issues
www.database.ittoolbox.com/topics Database Security
www.databasesecurity.com Database Security
www.smckearney.com/hncdb/notes/lec.security.pdf Database Security

OSD06-NC4 TITLE: Kernel-mode Software Protection Vulnerability Assessment and Rootkit Reverse Engineering Tool Development

TECHNOLOGY AREAS: Information Systems, Sensors, Electronics, Battlespace

OBJECTIVE: Develop an advanced tool that will allow a vulnerability assessment of kernel-mode software protection solutions and the reverse engineering of kernel-mode rootkit technology.

DESCRIPTION: The Anti-Tamper--Software Protection Initiative (AT-SPI) Technology Office is charged with preventing piracy, alteration, and reverse engineering of critical national security software and data. Software protection involves developing a defense-in-depth strategy using out-of-band defensive technologies that complement traditional in-band information assurance defenses, such as network firewalls and operating system access controls. In addition, AT-SPI requires a strategy that balances the need for application security with the need to have a minimal impact on performance and adoptability of the protection solution by authorized end-users. To meet this need, AT-SPI is currently performing research in kernel-mode software protection technology. This technology utilizes, in-part, rootkit-like methods that provide anti-piracy and anti-reverse engineering protection to critical software applications. The purpose of this research is to develop software protection technologies that are inaccessible to the attacker, increasing the cost to reverse engineer, while at the same time lowering the cost to protect the software.

Rootkit technology is also being used by malicious individuals and organizations. Using a rootkit, an adversary that acquires administrative access to a computer system can remain hidden from system administrators in order to exploit the computer system or network in which that computer resides. Kernel-mode tools can be used for reverse engineering and analysis of rootkits and other malicious kernel-level code that have not previously been observed in the wild. These tools could be used in order to better understand the tactics and techniques of malicious users so as to develop better detection capabilities and counter-measures to those attacks.

AT-SPI is interested in developing tools for kernel-mode rootkit reverse engineering for the purposes of information assurance, while simultaneously providing technology to enhance the protection strength of kernel-mode based software protection. Two use-cases that this research will address are (1) reverse engineering of kernel-level rootkits or kernel-level software protections prior to installation (e.g. an analysis of a kernel module prior to loading) and (2) reverse engineering of a kernel-level rootkit or kernel-level software protection after installation. In the first scenario, where kernel-level software protections are being installed, it is assumed that the attacker (possibly an insider) has complete control over the target machine. The second scenario corresponds to a situation where the protected software binary has already been installed and is now being attacked (possibly over a network). In this case, the attacker may not have physical access to the machine hardware except via a network connection. To measure the long-term effectiveness of such protection solutions, sophisticated user and kernel-level tools need to be developed. Tools of interest include, but are not limited to, kernel-level debuggers (local and remote), virtualization technology that supports kernel-level debugging, trace-based disassemblers, native x86 binary decompilers, and forensic tools that examine volatile memory and file systems. These tools can be used for both verification and validation of the software protection insertion process or as an attack tool for red teaming purposes. The proposed tools should advance the current state-of-the-art in each respective area.

PHASE I:
1) Develop a concept for a kernel-level reverse engineering tool that will allow the AT-SPI to assess the vulnerabilities of software protections and the reverse engineering of kernel-mode rootkits. This tool should be based on current offensive rootkits and defensive software protection technologies.
2) Design and build a prototype tool that proves the feasibility of the concept.
3) Perform a red team analysis of a kernel-based software protected binary executable or a rootkit reverse engineering analysis using the tool and deliver summary results of tool effectiveness.

PHASE II:
1) Based on the results from Phase I, refine and extend the design of the reverse engineering tool prototype to a complete toolset.
2) Develop the prototype toolset using industry best practices.
3) Perform a security penetration attack analysis on a kernel-mode protected binary executable using the toolset and deliver detailed results of attacks and protection effectiveness, to include attack trees, processes, times and similar data.

PHASE III DUAL-USE COMMERCIALIZATION: Tools and technologies for the protection of high-value software against reverse engineering would be marketable in both the DoD and commercial sectors. Computer applications where software vulnerabilities are a concern would benefit from these technologies. Tools and technologies that can examine malicious kernel-level code and rootkits would be a valuable asset to both the DoD and commercial security companies.

REFERENCES:
(1) Sandra Ring and Eric Cole, "Taking a Lesson from Stealthy Rootkits," IEEE Security and Privacy 1(4), 2004, pp. 38-45.
(2) Sandra Ring and Eric Cole, "Volatile Memory Computer Forensics to Detect Kernel-Level Compromise," Information and Communication Security, 6th International conference (ICICS), 2004.
(3) Greg Hoglund and James Butler, Rootkits: Subverting the Windows Kernel, Addison Wesley, 2005.
(4) Sherri Sparks and Jamie Butler, "Shadow Walker: Raising the Bar for Windows Rootkit Detection," Phrack Volume 63 (8), pp. 1-20.
(5) Ilo, "Advances in remote-exec Anti-Forensics," Phrack Inc, Vol. 0x0b, Issue 0x3f.
(6) Silvio Cesare, "Runtime kernel kmem patching," http://vx/netlux.org/lib/vsc07.html.
(7) Rainer Wichmann, "Linux Kernel Rootkits," http://la-samhna.de/library/rootkits/index.html.
(8) Christopher Kruegel, William Robertson and Giovanni Vigna, "Detecting Kernel-Level Rootkits Through Binary Analysis," 20th Annual Computer Security Applications Conference (ASSAC'04), pp. 91-100.

KEYWORDS: Rootkits, Linux Kernel Modules, Static Analysis, Dynamic Analysis, Debugger, Emulator, Execution Trace, Vulnerability Analysis, Reverse Engineering

OSD06-NC5 TITLE: Deobfuscating tools for the validation and verification of tamper-proofed software

TECHNOLOGY AREAS: Information Systems

OBJECTIVE: Develop deobfuscating tools and techniques to verify and validate software tamper-proofing techniques, and allow for the unobserved monitoring and detection of self-protecting and exfiltrating malicious code.

DESCRIPTION: As information resources have become network-centric, malicious code writers have become more sophisticated and are using a combination of different technologies to infiltrate networks and maintain access. For example, some spyware and viruses combine information egress or exfiltration capability with rootkit technology to hide files on the computer system in order to avoid detection. Spyware, viruses, and Trojans also contain a high-degree of self-protecting code, including encryption, obfuscation, anti-debugging, anti-disassembly, anti-emulation, polymorphic and metamorphic transformations in order to thwart reverse engineering and avoid detection. These self-protecting mechanisms can be applied at either the user or kernel level, increasing their strength and effectiveness. In addition, entire networks can become compromised by self-protecting malware before anti-virus and anti-spyware signatures can be built, released, and deployed by system administrators. Hence, tamper-proofed applications must be developed and general-purpose malicious behavior detectors must be embedded within the global information grid (GiG) to ensure the integrity and authenticity of the information system.

The Anti-Tamper-Software Protection Initiative (AT-SPI) Technology Office is charged with preventing piracy, alteration, and reverse engineering of critical national security software and data. Software protection involves developing a defense-in-depth strategy using out-of-band defensive technologies that complement traditional in-band information assurance defenses, such as network firewalls and operating system access controls. In order to provide information assurance in a network-centric environment, secure communication between hosts must be combined with application-centric security policies and products. These security products provide protection at the end-points of the network, such as remote computers that are connected to the Internet, where compromises from cyber threats most often occur. To provide network end-point protection, tamper-proofed software and monitoring systems must be developed, verified, validated, and deployed.

The difficultly with building tamper-proofed software applications is that tools do not currently exist to assess the effectiveness of the protection strength and ensure that additional vulnerabilities are not inadvertently introduced during the protection process. The very techniques that are designed to deter reverse engineering by adversaries also prevent the analysis, validation, verification, and monitoring of those same applications. This is particularly true when tamper-proofing is inserted directly into the binary executable. Currently, automated binary insertion techniques have no way to be validated or verified, since defensive protection networks embedded within the executable are designed to defeat analysis techniques and reverse engineering. Static analysis of binary executables must overcome anti-disassembly and anti-decompilation defenses before these tools can be used. Dynamic analysis must circumvent anti-debugging, anti-emulation, self-checksumming, and more targeted defenses before performing runtime analysis on the tamper-proofed or self-protecting binary. In most cases, the entire executable must be disassembled to provide whole program understanding and interrogation. In addition, compiler optimizations can remove or diminish the effectiveness of certain forms of protection, requiring analysis at the binary level subsequent to the compilation and linking phase in the software protection lifecycle.

The AT-SPI Technology Office is interested in innovative solutions to developing tools for verification and validation of tamper-proofed applications. The main requirement of the toolsuite is the ability to bypass or deobfuscate certain defensive features in tamper-proofed applications in order to perform the above functions. AT-SPI anticipates that the research and development effort under this STTR will form a foundation for several commercial products in the information assurance and software tamper-proofing arena, including

1. Automated tools and techniques that circumvent self-protecting malicious code and allow signatures to be automatically produced for runtime monitoring and detection systems.
2. Accurate round-trip disassemblers and decompilers that allow the insertion of tamper-proofed code within a binary executable. Tools such as deobfuscating disassemblers and decompilers can be used as part of an automated tamper-proofing insertion product, where tamper-proofing can be inserted in assembly language, intermediate-language, or high-level source code, depending on the level of code abstraction produced by the tool.
3. Runtime detection of self-protecting and exfiltrating malicious code. AT-SPI is interested in virtualization techniques such as process and system emulators that can bypass certain forms of self-protections used by malware and tamper-proofed software. In addition, dynamic translators and rewriters can be used to instrument or modify malicious program behavior at runtime.
4. Behavioral monitoring of tamper-proofed applications at runtime. Program understanding tools that attempt to statically model malicious behavior from binary executables must first circumvent the tamper-proofing and/or self-protecting malicious code embedded within the application.

AT-SPI is interested in techniques and tools that will assist in the verification and validation of tamper-proofed binary executables as well as the analysis and monitoring of self-protecting malicious code. Tools of interest include, but are not limited to, deobfuscating disassemblers, robust decompilers, virtualization technology, stealthy debuggers, dynamic binary translators, binary rewriters, and verifiable tamper-proofing binary insertion tools.

PHASE I:
1) Develop a concept for a deobfuscating tool based on current self-protecting malicious code and defensive software tamper-proofing protection technologies.
2) Design and build a minimal prototype that proves the feasibility of the concept. To minimize the development effort during Phase I, the leveraging of open source code or publicly available tools is acceptable.
3) Demonstrate the improvements to tamper-proofing and/or the verification and validation of tamper-proofed code offered by the prototype tool.

PHASE II:
1) Based on the results from Phase I, refine and extend the design of the verification and validation tool prototype to a complete toolset.
2) Develop the prototype toolset using industry best practices.

3) Perform verification and validation (V&V) analysis of a tamper-proofed binary executable using the toolset and deliver detailed results of methodology, protection effectiveness, and resistance to exploitation.

PHASE III DUAL-USE COMMERCIALIZATION: Software tamper-proofing is critical to protecting information security in both the military and commercial sectors. Military applications include ensuring authentication and integrity of command, control, and communication centers and the protection of high performance computing centers.

Commercial applications, where software vulnerabilities are a concern, would benefit from these tamper-proofing technologies. The automation of malicious code detection and signature generation could be integrated with network intrusion detection systems. Deobfuscating disassemblers, robust decompilers, virtualization technologies, and stealthy debuggers would be beneficial to both the DoD and commercial security companies, such as the anti-virus community. Vulnerability assessment tools would be of benefit to both the DoD and commercial companies developing digital rights management (DRM) software.

REFERENCES:

(1) Paul F. Robert, “Spyware Danger Meets Rootkit Stealth,” http://www.eweek.com/article2/0,1895,1829744,00.asp, June 20th, 2005.
(2) Bill Brenner, “Myfip’s Titan Rain connection,” http://searchsecurity.techtarget.com/originalContent/0,289142,sid14_gci1120855,00.html, August 31, 2005.
(3) (2) Peter Szor, The Art of Computer Virus Research and Defense, Addison Wesley, 2005.
(4) Frederic Perriot, Peter Ferrie, and Peter Szor, "Striking Similarities," Virus Bulletin, May 2002, pp. 4-6.
(5) (4) Peter Szor and Peter Ferrie, "Hunting for Metamorphic," http://enterprisesecurity.symantec.com/PDF/metamorphic.pdf
(6) Amit Vasudevan and Ramesh Yerraballi, “Stealth Breakpoints,” Proceedings of the 21st Annual Computer Security Applications Conference, 2005 (ACSAC 2005).
(7) Mike Van Emmerik and Trent Waddington, "Using a Decompiler for Real-World Source Recovery," IEEE 11th Working Conference on Reverse Engineering (WCRE), 2004, pp. 27-36.
(8) Sharath K Udupa, Saumya K. Debray, and Matias Madou, "Deobfuscation: Reverse Engineering Obfuscated Code," IEEE 12th Working Conference on Reverse Engineering (WCRE), 2005.
(9) Benjamin Schwarz, Saumya Debray, Gregory Andrews, "Disassembly of Executable Code Revisited," IEEE 9th Working Conference on Reverse Engineering (WCRE), 2002.
(10) Christopher Krugel, William Robertson, Fredrick Valeur, and Giovanni Vigna, "Static Disassembly of Obfuscated Binaries," 13th USENIX Security Symposium, pp. 255-270 of the Proceedings.

KEYWORDS: Tamper-proofing, Vulnerability Assessment, Reverse Engineering, Disassemblers, Decompilers, Debuggers, Dynamic Translators, Static Analysis, Dynamic Analysis

OSD06-SP1 TITLE: Error-handling paths and policies analysis

TECHNOLOGY AREAS: Information Systems

OBJECTIVE: Develop tools to analyze systems for potential error-handling paths and policies

DESCRIPTION: The reliability of software can be adversely impacted by the error-handling paths and policies established in the operating kernel and applications. In some instances, those paths and policies are incomplete in that they fail to anticipate reasonably foreseeable conditions within the software or system where erroneous inputs or states could occur. In other instances, the error handling policy fails to reflect context-specific (system-level or environmental) error conditions where a bound on the software or state is determined by external factors rather than within the software itself.

Approaches and tools to analyze the existence, completeness and adequacy of error-handling policies and paths would improve our ability to assess overall system quality and estimate reliability. They could also be used in the presence of reused or off-the-shelf software to allow a more in depth understand of the error-handling capabilities of software that assesses the potential impacts that software would have on the overall system.

PHASE I: Review existing approaches in defining and implementing error-handling paths and policies. Assess how error-handling is incorporated into software and software-based systems and methods for automatically determining error-handling paths and policies from source, object or binary code.

PHASE II: Develop new or revised tools to enable the automated or facilitate the determination and effectiveness of error-handling paths and policies.

PHASE III/DUAL USE COMMERCIALIZATION: Personal and commercial computing increasingly relies the integration of software applications that do not have detailed documentation on error-handling and have not been run within the system under development. Capabilities to assess the error-handling capabilities of the individual components and the end system will allow developers of personal and commercial applications to understand the extent to which error-handling has been achieved within their program or system. These developers will then be able to assess the adequacy of the resulting system in error-handling and, where necessary, make improvements.

REFERENCES:
[Kropp98] Kropp, Nathan P.; Koopman, Philip J.; Siewiorek, Daniel P., "Automated robustness testing of off-the-shelf software components." Twenty-Eighth Annual International Symposium on Fault-Tolerant Computing, June 1998, p. 230-239.

KEYWORDS: Error-handling, Software Reliability, Systems Engineering

OSD06-SP2 TITLE: <u>Security Escorts for Not-Yet-Trusted Software</u>

TECHNOLOGY AREAS: Information Systems

OBJECTIVE: Develop security escorts to accompany and monitor less than trustworthy software

DESCRIPTION: With the rapid release of new software applications, builds and patches for existing applications, and increased mobility of software across networks, more of our systems will run software applications that may not have a full pedigree of evaluation and testing to ensure that they are free from exploits and malware. Additionally, any previous certifications are invalidated at the first update. However, it isn't practical to exclude critical functionality from our systems simply due to a lack of pedigree. One technique for allowing untrusted functions to run on a system is to establish a constrained environment, or sandbox, that monitors the activities of software and limits access to the rest of the system such as Security Enhance Linux [2].

The goal of this topic is to develop security escorts to follow and monitor actions of not yet fully trusted software (e.g., possibly attached to packets) to report but not restrict full use of applications. Note that this will also require developing tools to perform automated analysis of those reports. Previous research on "wrappers," audit reduction, and network monitoring may support this work. Also note that the telecommunications community refers to "software auditing" components that have a somewhat similar concept, though they are usually used to support dependability not information assurance.

Existing or emerging approaches such as sandboxing or partitioning micro-kernels provide point solutions but do not address building systems with these security features, don't allow for run-time access in networks, nor do they provide monitoring capabilities for activity awareness to users and administrators.

PHASE I: Review existing approaches in managing the run-time activities of untrusted applications to assess current capabilities to limit, monitor and mitigate the activities of untrusted applications.

PHASE II: Develop new software and system tools that will allow untrusted applications to safely execute in a way that maximizes functionality but minimizes security risks. These applications may be installed by a user, administrator or enter the system via a network. These tools are expected to build on existing work in sandboxing and isolation but will include new monitoring features that provides users and administrators with activities of untrusted code so that appropriate manual or automated responsive actions can be taken when necessary.

PHASE III/DUAL USE COMMERCIALIZATION: Personal and commercial computing increasingly relies on the installation and updating of software as well as the use of networks and the Internet to install and update new applications. Techniques developed for DoD would increase the ability to rapidly install and update those systems without compromising critical data or functionality for individuals or companies as well.

REFERENCES:
[1] Secure Isolation and Migration of Untrusted Legacy Applications
Shaya Potter Jason Nieh Dinesh Subhraveti
Computer Science Department
Columbia University
spotter, nieh, dinesh@cs.columbia.edu
Columbia University Technical Report CUCS-005-04, January 2004

[2] Security-Enhanced Linux http://www.nsa.gov/selinux/

KEYWORDS: Security, Software Development, Systems Engineering, Software Assurance

OSD06-SP3 TITLE: <u>Assessing Interoperability Through Cross-Domain Protocol Compatibility Analysis</u>

TECHNOLOGY AREAS: Information Systems

OBJECTIVE: Develop tools to assess compatibility of data exchange mechanisms and associated application interactions between dissimilar domains.

DESCRIPTION: The explosive growth of networked computer systems has increased the opportunities for previously unrelated or weakly interdependent domains to interact. Often, these new interactions create opportunities for improved functionality to users but our ability to assess the feasibility of the new interactions is accomplished by manually reviewing the various protocols, implementations of protocols, and application-level interactions. DoD’s emphasis on the Global Information Grid and Net-Centric Enterprise Services will increase the likelihood that previously unrelated domains will interact in unforeseen ways.

Efforts in this area will develop tools to analyze compatibility between suites of protocols, implementations of protocols, and the associated applications activities to determine the level of compatibility and ability to interoperate between computing domains. Examples include the integration of data between combat system elements, such as target data from Sonar, Electronic Warfare, Navigation and Imaging Systems or the integration of supply consumption data from logistics functions with a combat system with supply ordering functions in an Enterprise Resource Planning system. As net-centric operations grow, unanticipated opportunities to collaborate will emerge.

PHASE I: Review existing approaches in defining protocols, ensuring protocol implementations are consistent, and determining the applications activities necessary to use the protocols. Assess research opportunities for and feasibility of building cross-domain protocol compatibility tools.

PHASE II: Develop new or revised approaches for assessing protocols, their implementations, and determining the compatibility in application processes needed to drive and exploit those protocols.

PHASE III/DUAL USE COMMERCIALIZATION: Personal and commercial computing increasingly relies the interaction of networked applications through protocols for communications and data exchange. Improved capabilities in assessing the compatibility will aid in assessing and enhancing interoperability for internet commerce in new domains and data fusion from disparate sources such as assessing economic health by examining purchasing trends across the internet.

KEYWORDS: Interoperability, Net-centric Operations, Systems Engineering

OSD06-SP4 TITLE: Software System Reliability Analysis

TECHNOLOGY AREAS: Information Systems

OBJECTIVE: Provide a methodology, tools and techniques for evaluation and estimation of software-based embedded systems reliability.

DESCRIPTION: Establishing the reliability of embedded software-based systems is increasingly important as the Department of Defense's dependency of software grows along with the size and complexity of programs. Currently, no theory exists that adequately explains the behavior of software systems reliability as software programs are inherently deterministic yet, in real world situations exhibit stochastic behavior as they interact with users and the environment. To date, reliability estimation for these systems is done either by extending concepts from hardware reliability (example – F-22 avionics stability Mean Time Between Avionics Anomalies) or correlating software readiness with defects and operational profiles (Musa profiles).

Performers in this topic will be expected to update current approaches or develop new ideas that adequately characterize software systems reliability and can provide mechanisms to predict system reliability from software development artifacts (specifications, code, test results) and hardware characterizations. These updated or new approaches will be the basis for a framework to express software system reliability concepts, tools/techniques to estimate reliability, and tools/techniques for evaluating the reliability of large software systems through bench and field testing. Specific focus on high criticality embedded systems is of primary interest although theories should address software-based systems in general, including those relying heavily on networks for interaction with other systems.

PHASE I: Review existing bodies of work in software reliability and computer science theories to assess current capabilities to understand, express, estimate and evaluate system reliability. Assess opportunities to conduct further research or extend theories of computer and information systems, systems engineering, reliability estimation into software system reliability.

PHASE II: Revise or develop approaches for understanding, and expressing embedded software system reliability. Develop tools and techniques for estimating and evaluating embedded software system reliability.

PHASE III/DUAL USE COMMERCIALIZATION: Embedded system reliability estimation and evaluation is of interest to the medical industry, commercial avionics industry, automotive industry to name a few. New approaches for DoD applications would equally apply in these areas as well. Software and software-based system reliability is of growing interest as more of our daily activities are enabled by software.

REFERENCES:
[1] IEEE Standard Dictionary of Software Engineering Terms

[2] Software Metrics and Reliability, by Dr. Linda Rosenberg, Ted Hammer, and Jack Shaw, November 1998, http://satc.gsfc.nasa.gov/support/ISSRE_NOV98/software_metrics_and_reliability.html

[3] Musa, J.D., A. Iannino and K. Okumoto, Software Reliability: Measurement, Prediction, Application, Professional Edition: Software Engineering Series, McGraw–Hill, New York, NY., 1990.

KEYWORDS: Reliability, Software Development, Systems Engineering